AVEVA

Integration Guide

DETAILED PRODUCTION SCHEDULING
WITH REAL-TIME DATA

Simio
Forward Thinking

Table of Contents

Simio
Forward Thinking

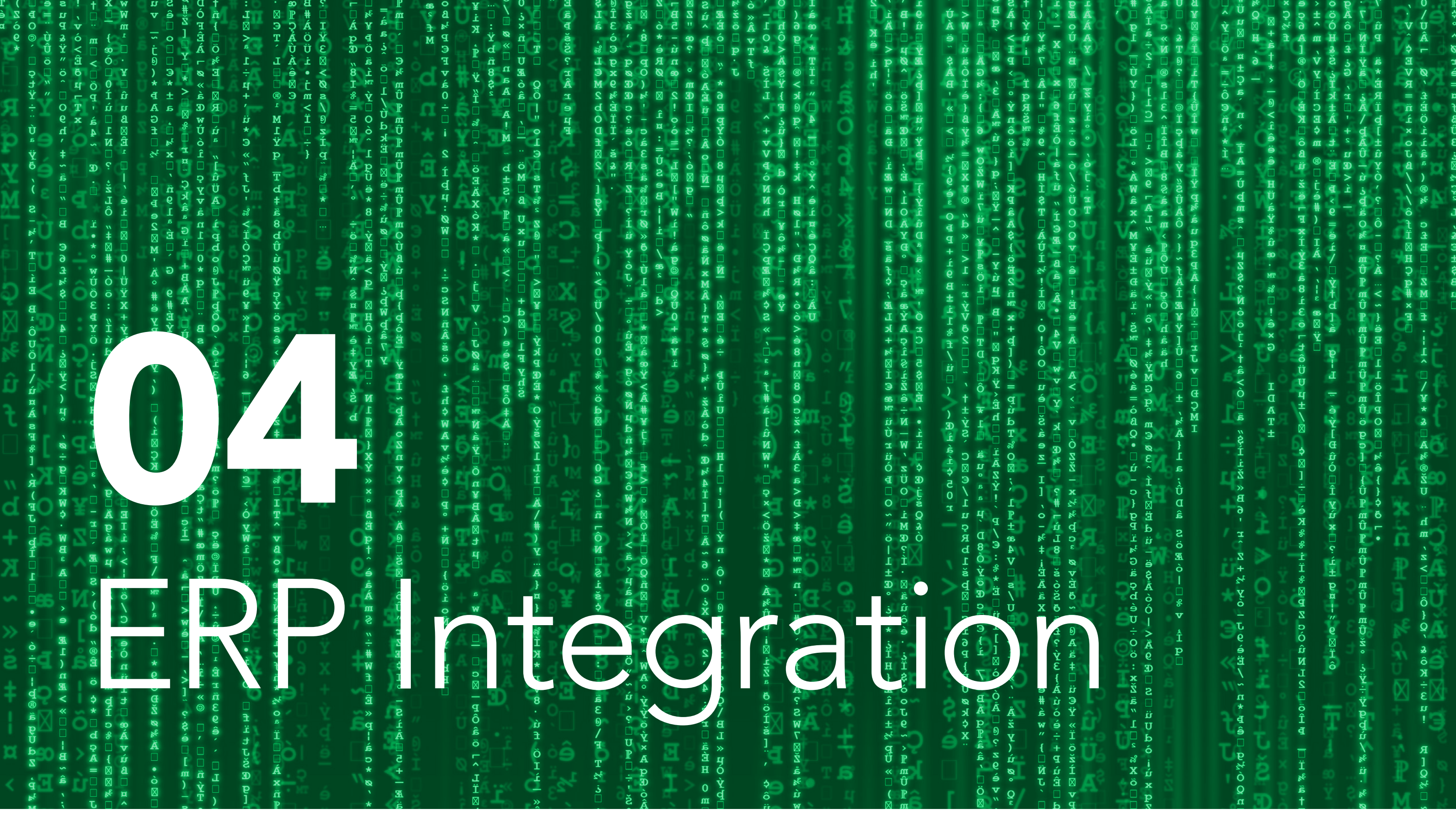
03 Overview

ORDERS, ROUTINGS, INVENTORY, SCHEDULE DATE, STATUS, CONFIRMATIONS, ETC.

Welcome to the world of simulation. Simulation provides a unique way to examine the future and make intelligent decisions based on what you learn. While simulation technology has been around for decades, it is still rapidly evolving. Advances in object-oriented approaches provide rapid modeling and the flexibility to model complex systems that could not be modeled just a few years ago. And integrated 3D animation makes the creation of compelling 3D visualizations easy, and this in turn helps assure more robust, understandable models and better communication with stakeholders.

Simio extends that base technology to provide a unique solution for planning and scheduling. Simio provides AVEVA MES with Finite Capacity Scheduling and Advanced Planning and Scheduling functionality. Simio's patented approach to planning and scheduling is first to automate comparing a deterministic schedule from MRP with a variation adjusted schedule to let you effectively deal with breakdowns, unplanned events and material shortages. You can then re-schedule resources to acceptable levels of risk. Data is brought into memory for fast execution. While Simio takes full advantage of your existing data to quickly and automatically build you scheduling model, that base model can also be enhanced as needed to accurately capture complexity unique to your system.

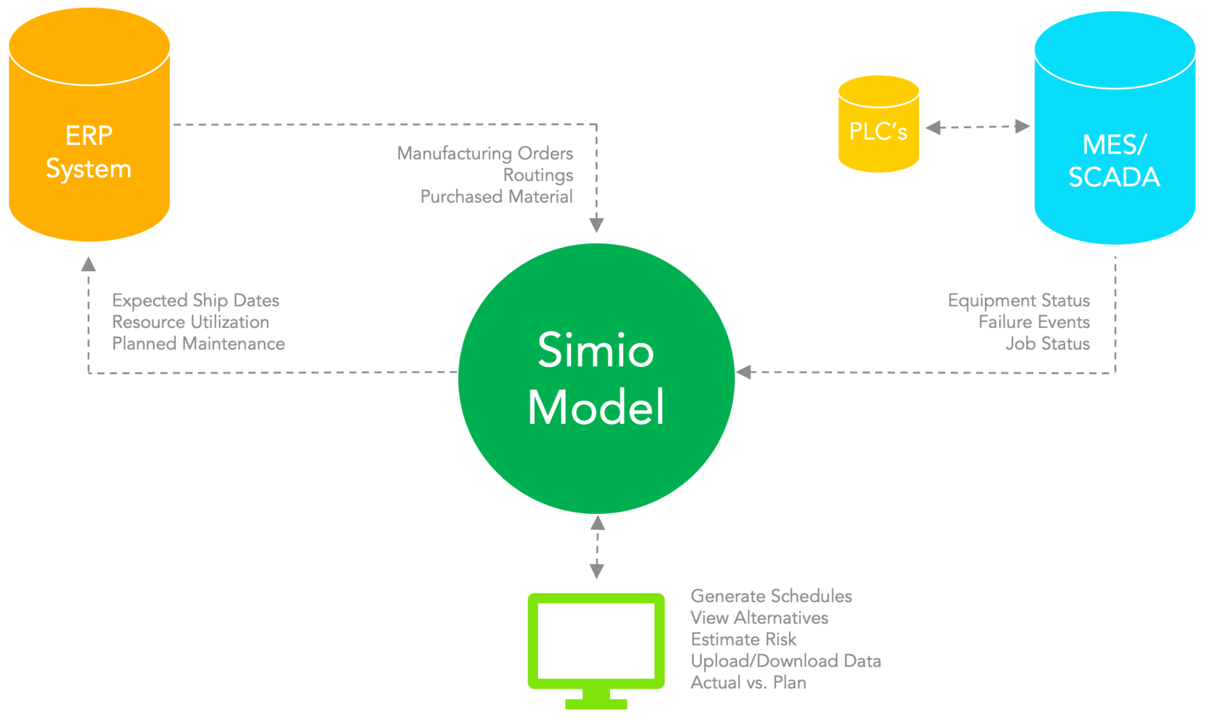
The team of architects behind Simio have been leaders in simulation since the early 1980's, playing key roles in the design and development of four previous market-leading products. Simio is the result of this team applying their collective 180 years of simulation experience and using the very latest in technology and development techniques to create a new generation of simulation problem solving capability.



04 ERP Integration

The Simio Production Scheduling model is typically integrated with the existing ERP and MES/SCADA system to provide the data on the current status of the system and the actual jobs to be processed through the system. This data is provided to Simio in the form of relational data sets that are imported and held in memory by Simio for fast execution.

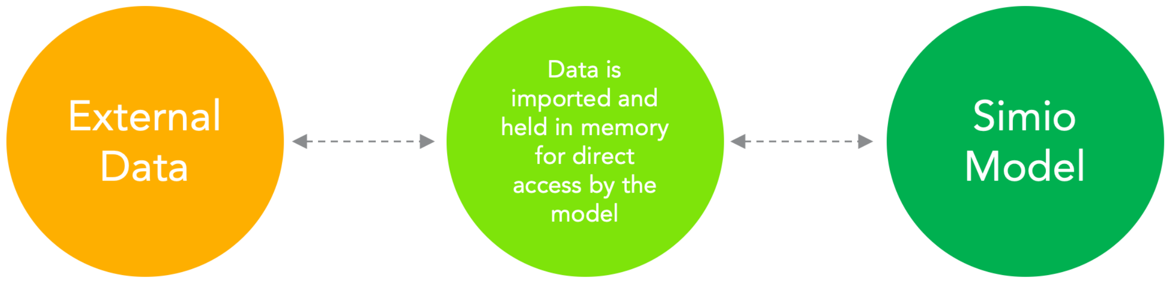
These data sets typically contain data such as a list of jobs to be processed, a bill of material for each job, job routings including setup and processing times, purchased material, etc., along with the status of all jobs in the system at the start of the planning period. Simio does not have a fixed data schema – the data sets used by Simio are configured as needed to match the form of the external data. The flow of jobs as defined by the data sets is then simulated by Simio using the model to generate a detailed schedule.



Integration between Simio Production Scheduling and AVEVA MES

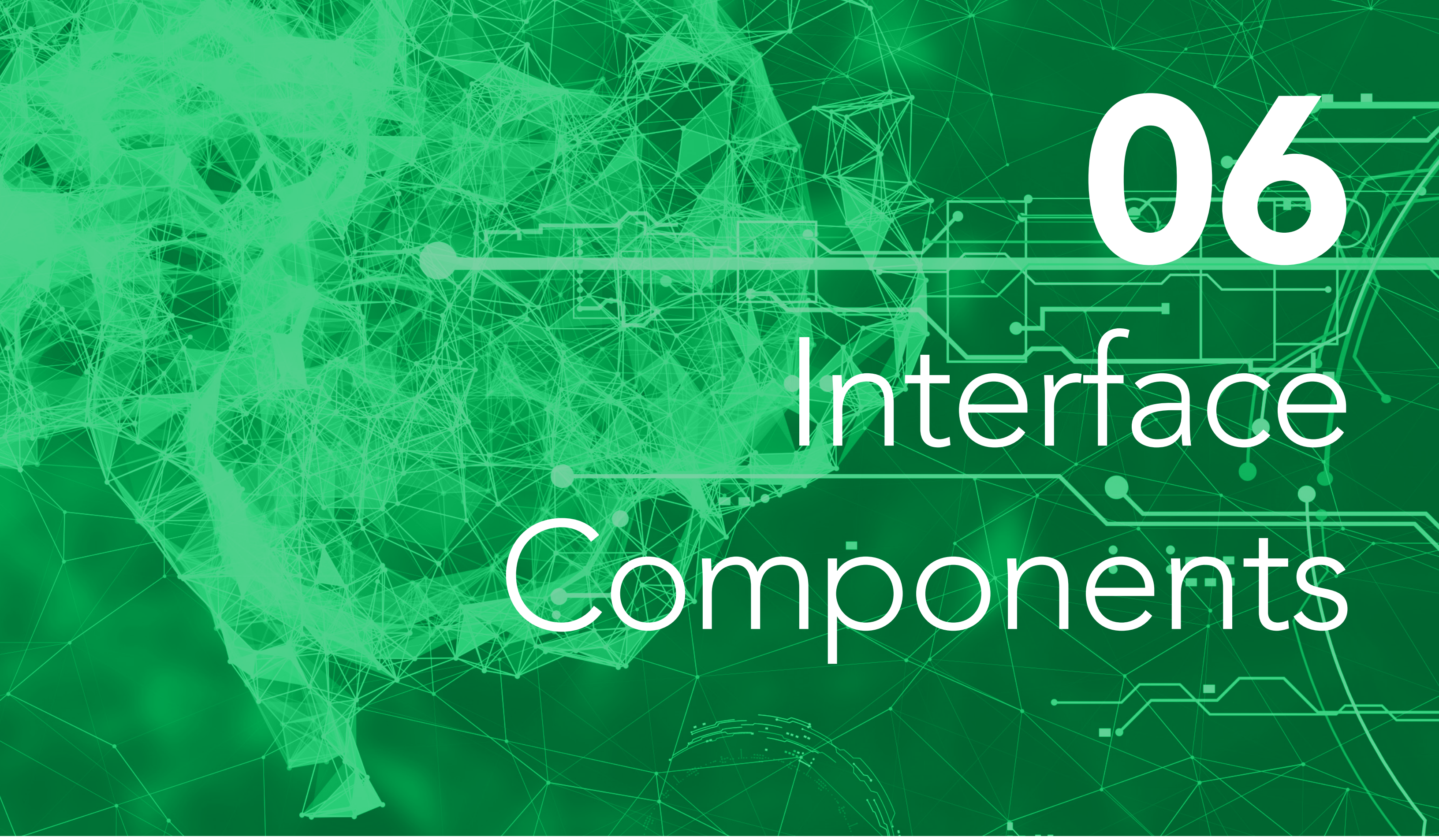
This interface has been built to integrate data between Simio Production Scheduling and AVEVA MES to enable real-time event based scheduling. Utilizing Simio's simulation-based scheduling capabilities, a model can be developed to depict the resource and material constraints within a manufacturing system. Then, interfacing Simio to the AVEVA MES enables the scheduling model to be run near real-time. This mean as events happen on the factory floor, the production schedule can be re-run quickly to re-route orders around possible bottlenecks.

The integration employs pulling of data from database views and stored procedures of the MES tables to get data into the model. The AVEVA MES as Web API Data Connector to call the AVEVA Web API to pull this data from the AVEVA MES data model. Inventory, work orders, routings, bill-of-materials, order status and machine status are all captured through this inbound integration to pull the data for creating the production schedule.



External Data
Data is imported and held in memory for direct access by the model
Simio Model

Once the schedule is generated, the scheduled is updated in the AVEVA MES using the AVEVA MES API. The Export Schedule to MES also uses a Web API Data Connector and the AVEVA Web API.



06 Interface Components

There are a number of components that are used for this integration. These components can be downloaded using the following URL:

https://github.com/SimioLLC/AVEVAMES

Unzip the file. The contents are as follows:

AVEVAMESMSSQL2019Backup
A Microsoft SQL Server 2019 database backup is provided to recreate the demo if someone has AVEVA MES 2017 R2.

DatabaseScripts
SQL views and stored procedures used to pull the MES data.

DemoModels
There are 2 completed scheduling models. One model shows the integration with AVEVAMES and the other shows the integration with InBatch.

Documents
This folder contains 2 documents. The first this SimioAPI-UsingTheSimioBackendDll.pdf that shows how to run Simio without the GUI. The MiddlewareExtensibility_Jan2013.pdf is a reference on how to trigger evens from the MES Middleware.

InBatchXMLStylesheets
This folder contains a sample InBatch recipe in BatchML format as well as 4 XML Stylesheets used to transform the BatchML into the format needed to import the data into the Simio tables.

Node-REDFlows
This folder contains the flow used to automatically regenerate the schedule based on MQTT Topic and Simio Portal Web API.

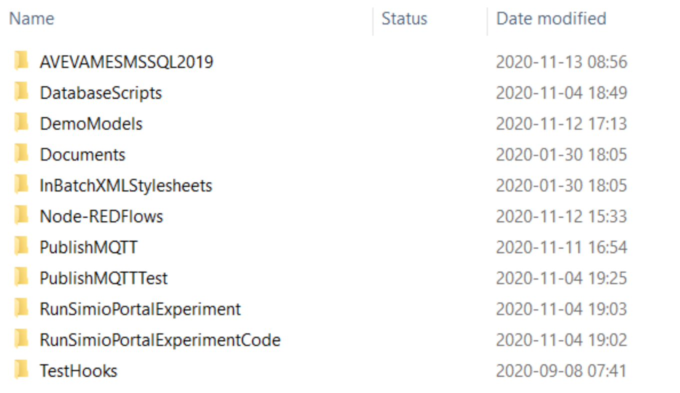
PublishMQTT
The folder contains the visual studio project that calls an MQTT topic called "mes/downtimeEvent" when configured to run via the AVEVAMES middleware. This is to be used with the Node-RedFlows.

PublishMQTTTest
Test Client to run PublishMQTT

RunSimioPortalExperiment
The folder contains the RunSimioPortalExperiment executable and supporting files to run Simio Portal Experiment using a Windows Service.

RunSimioPortalExperimentCode
The folder contains the visual studio project used to generated the RunSimioPortalExperiment executable.

TestHooks
The folder contains the visual studio project that produces a file when configured to run via the AVEVAMES middleware. This is to be used with the RunSimioPortalExperiment.



Name	Status	Date modified
AVEVAMESMSSQL2019		2020-11-13 08:56
DatabaseScripts		2020-11-04 18:49
DemoModels		2020-11-12 17:13
Documents		2020-01-30 18:05
InBatchXMLStylesheets		2020-01-30 18:05
Node-REDFlows		2020-11-12 15:33
PublishMQTT		2020-11-11 16:54
PublishMQTTTest		2020-11-04 19:25
RunSimioPortalExperiment		2020-11-04 19:03
RunSimioPortalExperimentCode		2020-11-04 19:02
TestHooks		2020-09-08 07:41

The DatabaseScript need to be run on the AVEVAMES database to setup the integration. The other folders are optional depending on the project. If needed, they will be referenced in other sections of this document.

08 Install Database Objects

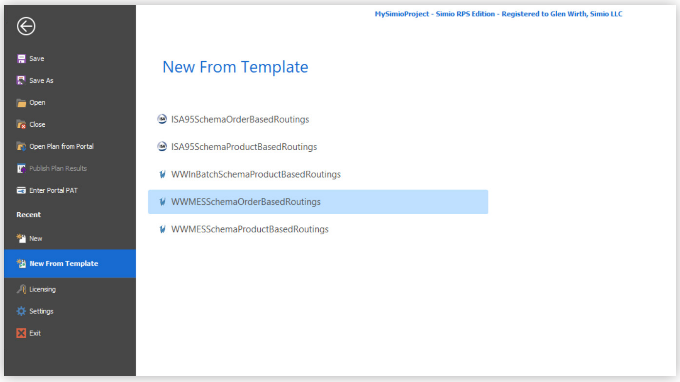
The database views and stored procedures are used to map the data between the AVEVA MES database and the Simio table. A database objects are in the SQL script provided. The scripts are to be run on the same database used by AVEVA MES.

Install these two scripts:

1. **DatabaseViews.sql**
   This script contains all the database views used in this integration.

2. **DatabaseStoredProcedures.sql**
   This script contains all the stored procedures used in this integration.

09 New Simio Model Based On Template

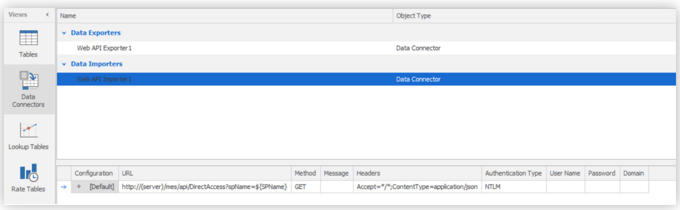

MySimioProject - Simio RPS Edition - Registered to Glen Wirth, Simio LLC

New From Template

- ISA95SchemaOrderBasedRoutings
- ISA95SchemaProductBasedRoutings
- WWInBatchSchemaProductBasedRoutings
- WWMESSchemaOrderBasedRoutings
- WWMESSchemaProductBasedRoutings

1. Open Simio
2. Select File "New From Template"
3. Select "WWMES Schema Order Based Routings"

This will create a new model that has a data schema, sub-classed object definitions, dashboards and reports defined. The new model can then easily be integrated with a existing AVEVAMES system that has order based routings.



Name	Object Type
Data Exporters	
Web API Exporter1	Data Connector
Data Importers	
Web API Importer1	Data Connector

Configuration	URL	Method	Message	Headers	Authentication Type	User Name	Password	Domain
+ [Default]	http://(server)/mes/api/DirectAccess?spName=${SPName}	GET		Accept=*/*;ContentType=application/json	NTLM			

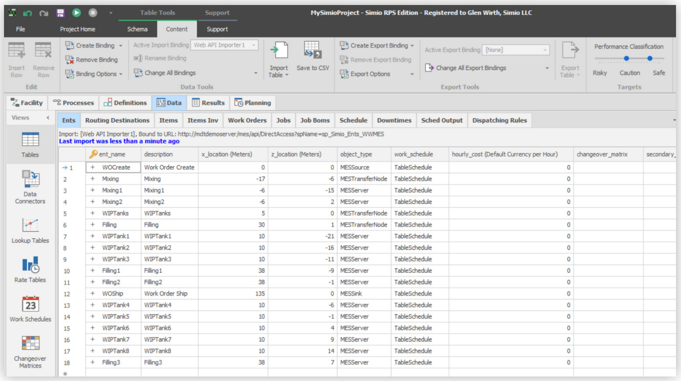
4. Select the Data tab and the Data Connectors.
5. Configure URL, UserName, Password and Domain.

SIMIO LLC AVEVA INTEGRATION GUIDE NEW SIMIO MODEL BASED ON TEMPLATE | **09**

10 Import Entities

In this step, we will import the entities (referred to as resources in Simio) from AVEVA MES into Simio.

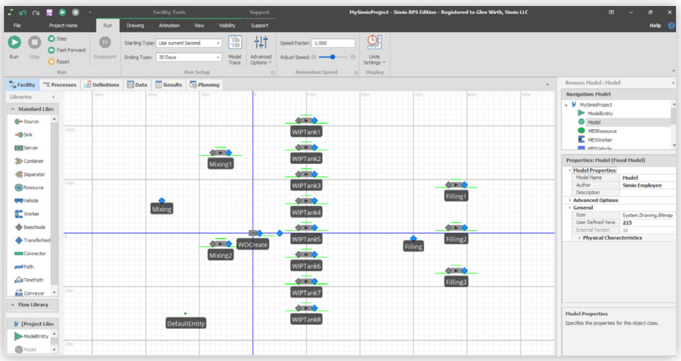
1. Select the DATA tab
2. Select the ENTS table
3. Select IMPORT table



Import: [Web API Importer1], Bound to URL: http://mdtdemoserver/mes/api/DirectAccess?spName=sp_Simio_Ents_WWMES
Last import was less than a minute ago

	ent_name	description	x_location (Meters)	z_location (Meters)	object_type	work_schedule	hourly_cost (Default Currency per Hour)	changeover_matrix	secondary_
1	WOCreate	Work Order Create	0	0	MESSource	TableSchedule	0		
2	Mixing	Mixing	-17	-6	MESTransferNode	TableSchedule	0		
3	Mixing1	Mixing1	-6	-15	MESServer	TableSchedule	0		
4	Mixing2	Mixing2	-6	2	MESServer	TableSchedule	0		
5	WIPTanks	WIPTanks	5	0	MESTransferNode	TableSchedule	0		
6	Filling	Filling	30	1	MESTransferNode	TableSchedule	0		
7	WIPTank1	WIPTank1	10	-21	MESServer	TableSchedule	0		
8	WIPTank2	WIPTank2	10	-16	MESServer	TableSchedule	0		
9	WIPTank3	WIPTank3	10	-11	MESServer	TableSchedule	0		
10	Filling1	Filling1	38	-9	MESServer	TableSchedule	0		
11	Filling2	Filling2	38	-1	MESServer	TableSchedule	0		
12	WOShip	Work Order Ship	135	0	MESSink	TableSchedule	0		
13	WIPTank4	WIPTank4	10	-6	MESServer	TableSchedule	0		
14	WIPTank5	WIPTank5	10	-1	MESServer	TableSchedule	0		
15	WIPTank6	WIPTank6	10	4	MESServer	TableSchedule	0		
16	WIPTank7	WIPTank7	10	9	MESServer	TableSchedule	0		
17	WIPTank8	WIPTank8	10	14	MESServer	TableSchedule	0		
18	Filling3	Filling3	38	7	MESServer	TableSchedule	0		

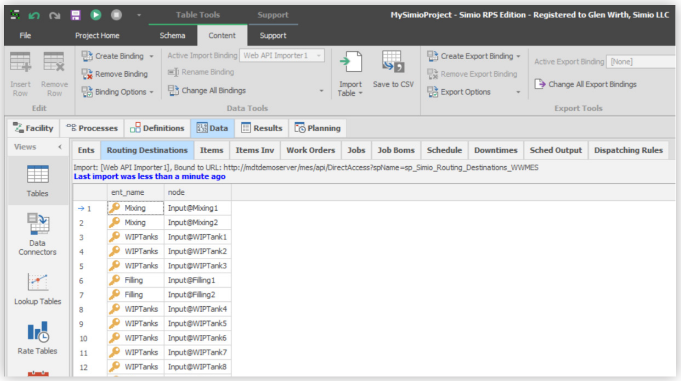
As the entities are added to the table, they are automatically added to the model. Select the 'Facility' tab. The entities are positioned at the location defined in the Ents table. If you choose to move the entities in the facility, the position will be updated in the Ents table.



Your Facility view is ready to use. It is good to save your model at this time. You can choose File…Save As to choose a different name for the model.

11 Import Product Definition and Order Management Tables

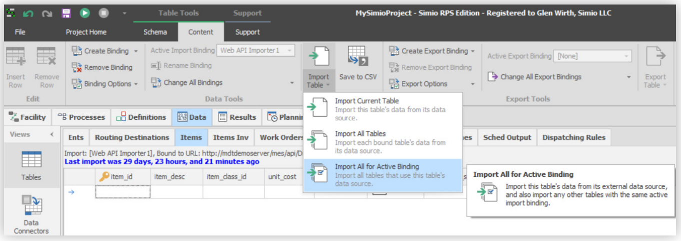
Follow the same database binding steps described in the IMPORT ENTITIES section to import the rest of the data. On each table that has an import binding, press the to import the data into the model.



Import: [Web API Importer1], Bound to URL: http://mdtdemoserver/mes/api/DirectAccess?spName=sp_Simio_Routing_Destinations_WWMES
Last import was less than a minute ago

	ent_name	node
→ 1	Mixing	Input@Mixing1
2	Mixing	Input@Mixing2
3	WIPTanks	Input@WIPTank1
4	WIPTanks	Input@WIPTank2
5	WIPTanks	Input@WIPTank3
6	Filling	Input@Filling1
7	Filling	Input@Filling2
8	WIPTanks	Input@WIPTank4
9	WIPTanks	Input@WIPTank5
10	WIPTanks	Input@WIPTank6
11	WIPTanks	Input@WIPTank7
12	WIPTanks	Input@WIPTank8

You can also press 'Import All for Active Binding' to import all the data into the model with a single button press.



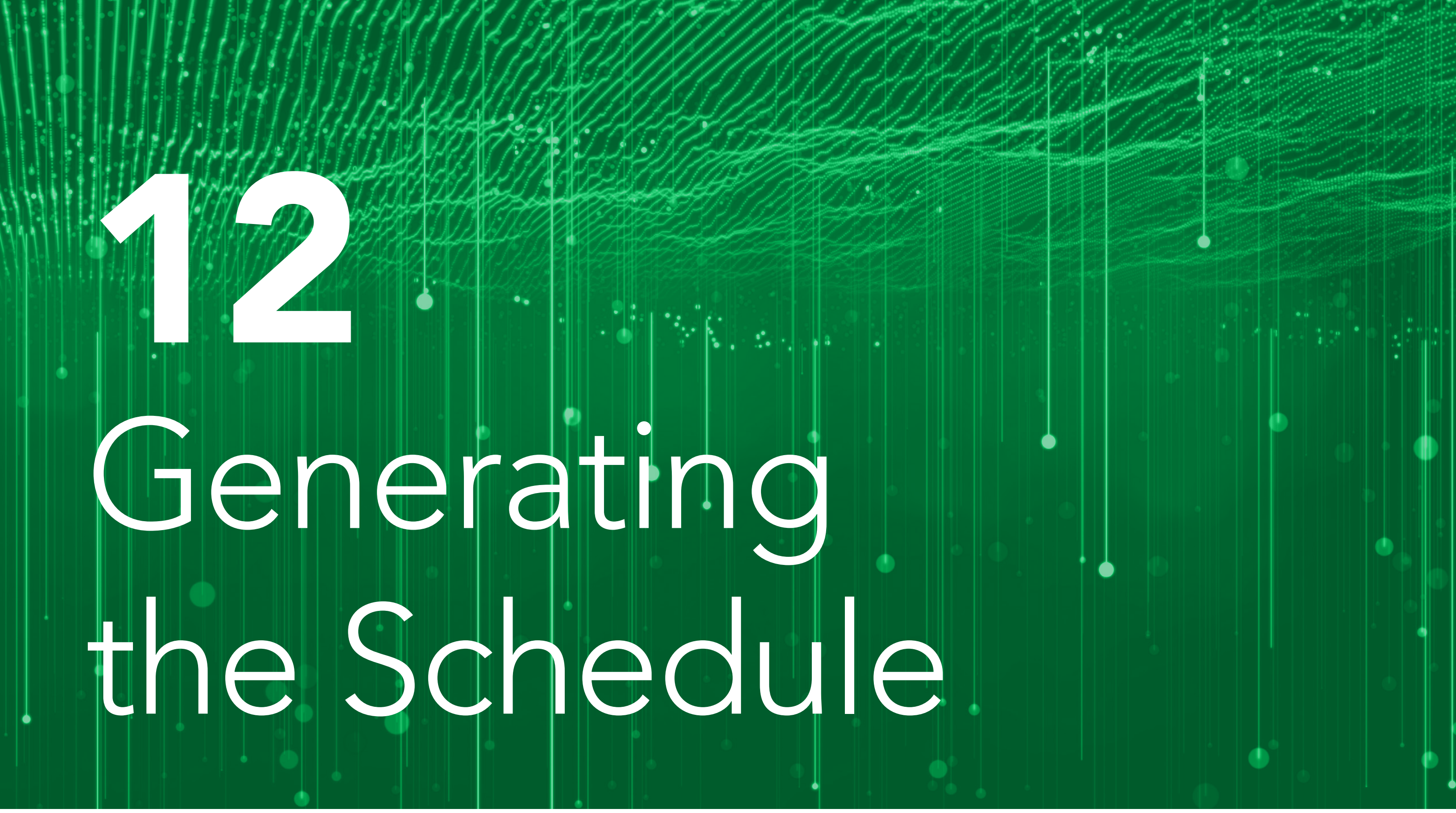
Import Current Table
Import this table's data from its data source.

Import All Tables
Import each bound table's data from its data source.

Import All for Active Binding
Import all tables that use this table's data source.

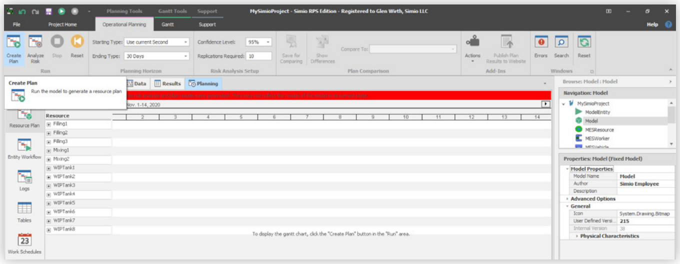
Import: [Web API Importer1], Bound to URL: http://mdtdemoserver/mes/api/D
Last import was 29 days, 23 hours, and 21 minutes ago

Import All for Active Binding
Import this table's data from its external data source, and also import any other tables with the same active import binding.

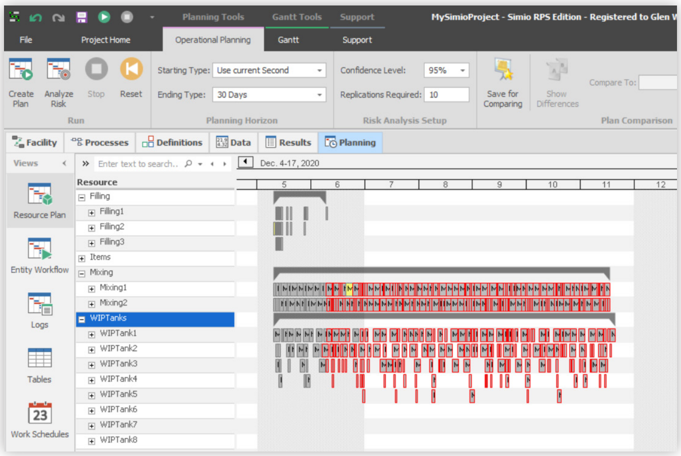


12 Generating the Schedule

Select the 'Planning' tab. Once selected, the 'Resource Plan" should be populated with the AVEVA MES Entities. From here, the schedule can be generated. By default, the work orders will be scheduled based on priority. To create the schedule, select "Create Plan".


Create Plan
Run the model to generate a resource plan
Resource Plan
Entity Workflow
Logs
Tables
Work Schedules
To display the gantt chart, click the "Create Plan" button in the "Run" area.

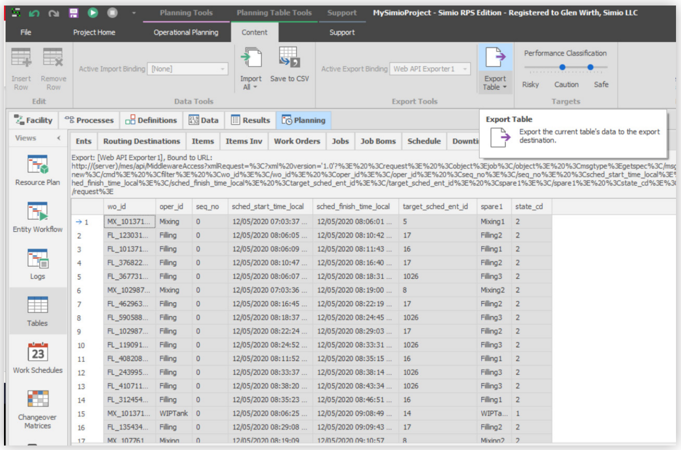
This will create the schedule. From the Gantt tab, select "All" to see the entire schedule.



Starting Type: Use current Second
Ending Type: 30 Days
Confidence Level: 95%
Replications Required: 10

13 Exporting Results

By default, the template is setup to automatically export the schedule back to AVEVA MES at the end of RunPlan. You can also do it manually by selecting the Tables from the Planning tab. Then, select the SchedOutput table and press 'Export Table'.



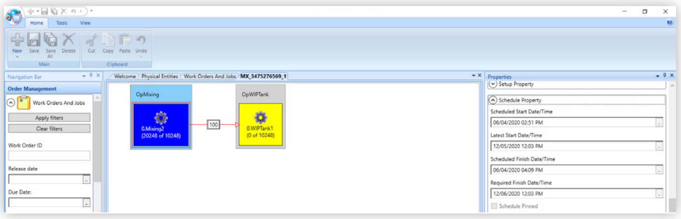
Export Table

Export the current table's data to the export destination.

Export: [Web API Exporter1], Bound to URL:
http://(server)/mes/api/MiddlewareAccess?xmlRequest=%3C?xml%20version='1.0'?%3E%20%3Crequest%3E%20%3Cobject%3Ejob%3C/object%3E%20%3Cmsgtype%3Egetspec%3C/msg
new%3C/cmd%3E%20%3Cfilter%3E%20%3Cwo_id%3E%3C/wo_id%3E%20%3Coper_id%3E%3C/oper_id%3E%20%3Cseq_no%3E%3C/seq_no%3E%20%3Csched_start_time_local%3E
hed_finish_time_local%3E%3C/sched_finish_time_local%3E%20%3Ctarget_sched_ent_id%3E%3C/target_sched_ent_id%3E%20%3Cspare1%3E%3C/spare1%3E%20%3Cstate_cd%3E%3
/request%3E

	wo_id	oper_id	seq_no	sched_start_time_local	sched_finish_time_local	target_sched_ent_id	spare1	state_cd
→ 1	MX_101371...	Mixing	0	12/05/2020 07:03:37 ...	12/05/2020 08:06:01 ...	5	Mixing1	2
2	FL_123031...	Filling	0	12/05/2020 08:06:05 ...	12/05/2020 08:10:42 ...	17	Filling2	2
3	FL_101371...	Filling	0	12/05/2020 08:06:09 ...	12/05/2020 08:11:43 ...	16	Filling1	2
4	FL_376822...	Filling	0	12/05/2020 08:10:47 ...	12/05/2020 08:16:40 ...	17	Filling2	2
5	FL_367731...	Filling	0	12/05/2020 08:06:07 ...	12/05/2020 08:18:31 ...	1026	Filling3	2
6	MX_102987...	Mixing	0	12/05/2020 07:03:36 ...	12/05/2020 08:19:00 ...	8	Mixing2	2
7	FL_462963...	Filling	0	12/05/2020 08:16:45 ...	12/05/2020 08:22:19 ...	17	Filling2	2
8	FL_590588...	Filling	0	12/05/2020 08:18:37 ...	12/05/2020 08:24:45 ...	1026	Filling3	2
9	FL_102987...	Filling	0	12/05/2020 08:22:24 ...	12/05/2020 08:29:03 ...	17	Filling2	2
10	FL_119091...	Filling	0	12/05/2020 08:24:52 ...	12/05/2020 08:33:31 ...	1026	Filling3	2
11	FL_408208...	Filling	0	12/05/2020 08:11:52 ...	12/05/2020 08:35:15 ...	16	Filling1	2
12	FL_243995...	Filling	0	12/05/2020 08:33:37 ...	12/05/2020 08:38:14 ...	1026	Filling3	2
13	FL_410711...	Filling	0	12/05/2020 08:38:20 ...	12/05/2020 08:43:34 ...	1026	Filling3	2
14	FL_312454...	Filling	0	12/05/2020 08:35:23 ...	12/05/2020 08:46:51 ...	16	Filling1	2
15	MX_101371...	WIPTank	0	12/05/2020 08:06:25 ...	12/05/2020 09:08:49 ...	14	WIPTa...	1
16	FL_135434...	Filling	0	12/05/2020 08:29:08 ...	12/05/2020 09:09:43 ...	17	Filling2	2
17	MX_107761	Mixing	0	12/05/2020 08:19:09	12/05/2020 09:10:57	8	Mixing2	2

In AVEVA MES, you will see that the jobs have been updated with the Simio scheduling information.



Scheduled Start Date/Time: 06/04/2020 02:51 PM

Latest Start Date/Time: 12/05/2020 12:03 PM

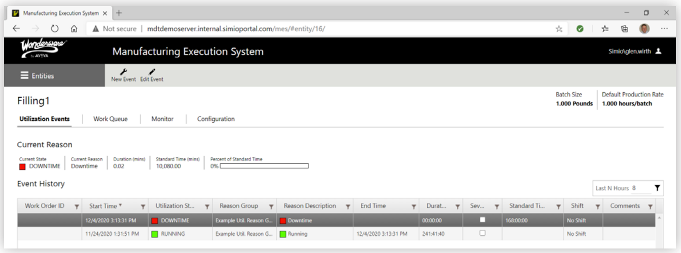
Scheduled Finish Date/Time: 06/04/2020 04:09 PM

Required Finish Date/Time: 12/06/2020 12:03 PM

☐ Schedule Pinned

14 Import Downtime

Add downtime on an entity in AVEVA MES Portal



Filling1

Batch Size: 1.000 Pounds | Default Production Rate: 1.000 hours/batch

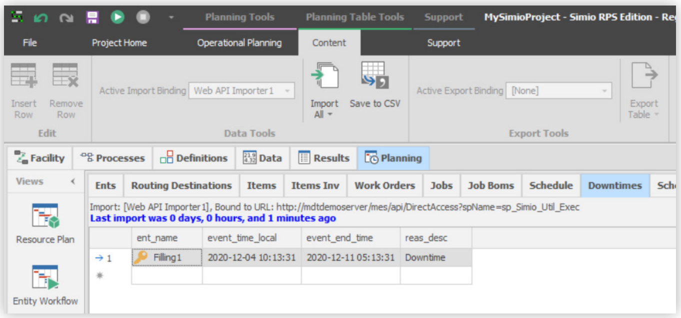
Current Reason

Current State	Current Reason	Duration (mins)	Standard Time (mins)	Percent of Standard Time
🟥 DOWNTIME	Downtime	0.02	10,080.00	0%

Event History

Work Order ID	Start Time	Utilization St...	Reason Group	Reason Description	End Time	Durat...	Sev...	Standard Ti...	Shift	Comments
	12/4/2020 3:13:31 PM	🟥 DOWNTIME	Example Util. Reason G...	🟥 Downtime		00:00:00		168:00:00	No Shift	
	11/24/2020 1:31:51 PM	🟩 RUNNING	Example Util. Reason G...	🟩 Running	12/4/2020 3:13:31 PM	241:41:40			No Shift	

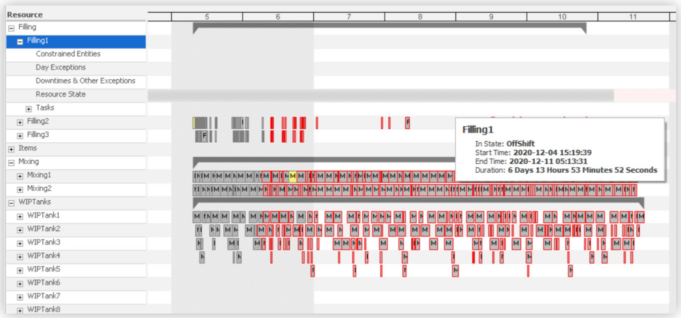
Within Simio, select the Tables from the Planning tab. Then, select the Downtimes table and press 'Import Table'.



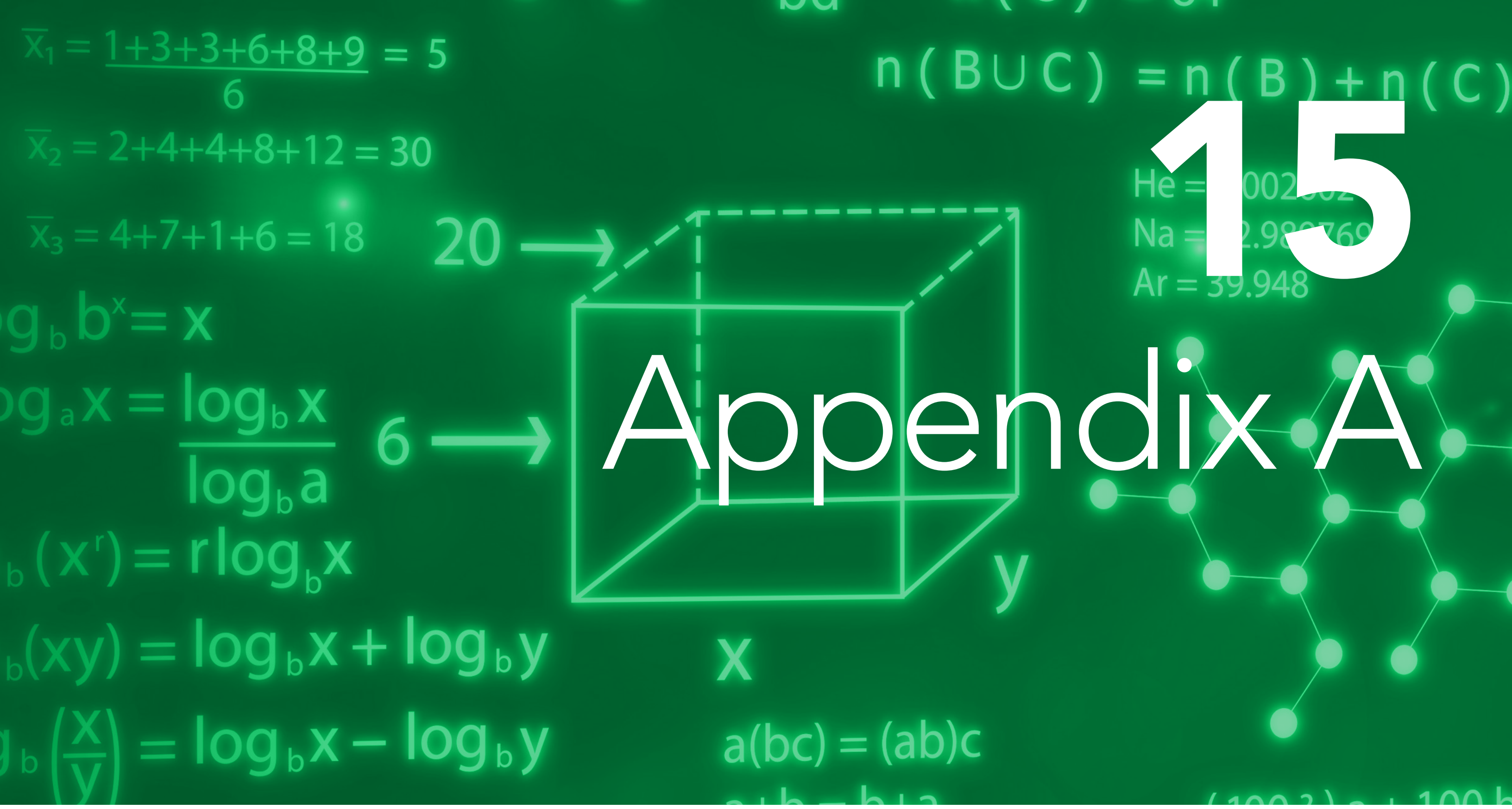
Import: [Web API Importer1], Bound to URL: http://mdtdemoserver/mes/api/DirectAccess?spName=sp_Simio_Util_Exec
Last import was 0 days, 0 hours, and 1 minutes ago

	ent_name	event_time_local	event_end_time	reas_desc
1	Filling1	2020-12-04 10:13:31	2020-12-11 05:13:31	Downtime

Re-plan around downtime. You will see that jobs have been shifted from Filler1 to the other mixer resources.



Filling1
In State: OffShift
Start Time: 2020-12-04 15:19:39
End Time: 2020-12-11 05:13:31
Duration: 6 Days 13 Hours 53 Minutes 52 Seconds

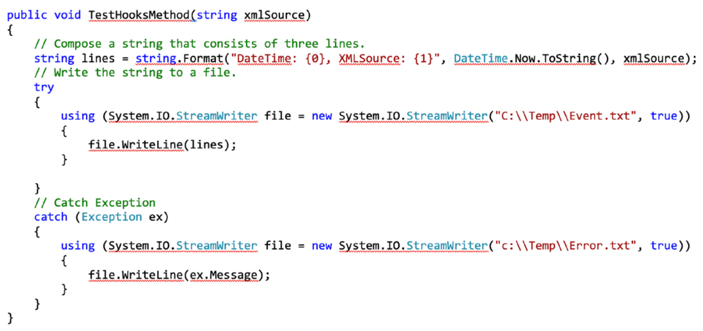


15 Appendix A

Automatic Reschedule Using TestHooks and RunSimioPortalExperiment

To automate reschedule, the AVEVA MES Middleware Extensibility Hooks and the Simio Portal Web API are used. The AVEVA MES Middleware Extensibility Hooks are used to capture the events in AVEVA MES. Once the events are captured, the RunSimioPortalExperiment service will see the event and automatically re-run the schedule using the Simio Portal Web API. After the scheduling run, the schedule will automatically be sent back to MES and published on Simio Portal.

To setup the AVEVA MES Middleware Extensibility Hooks, the 'Custom Assembly That Is Not in the GAC' example that start on page 26 of the MiddlewareExtensibility_Jan2013.pdf guide was referenced. The example was modified slightly to change the name of the output file and to capture any exceptions. The method used is:



```
public void TestHooksMethod(string xmlSource)
{
    // Compose a string that consists of three lines.
    string lines = string.Format("DateTime: {0}, XMLSource: {1}", DateTime.Now.ToString(), xmlSource);
    // Write the string to a file.
    try
    {
        using (System.IO.StreamWriter file = new System.IO.StreamWriter("C:\\Temp\\Event.txt", true))
        {
            file.WriteLine(lines);
        }

    }
    // Catch Exception
    catch (Exception ex)
    {
        using (System.IO.StreamWriter file = new System.IO.StreamWriter("c:\\Temp\\Error.txt", true))
        {
            file.WriteLine(ex.Message);
        }
    }
}
```

This assembly was compiled using the same structure defined in the 'Custom Assembly That Is Not in the GAC' example.

C:\Temp\TestHooks\bin\Debug\TestHooks.dll;TestHooksNamespace.TestHooksClass;TestHooksMethod

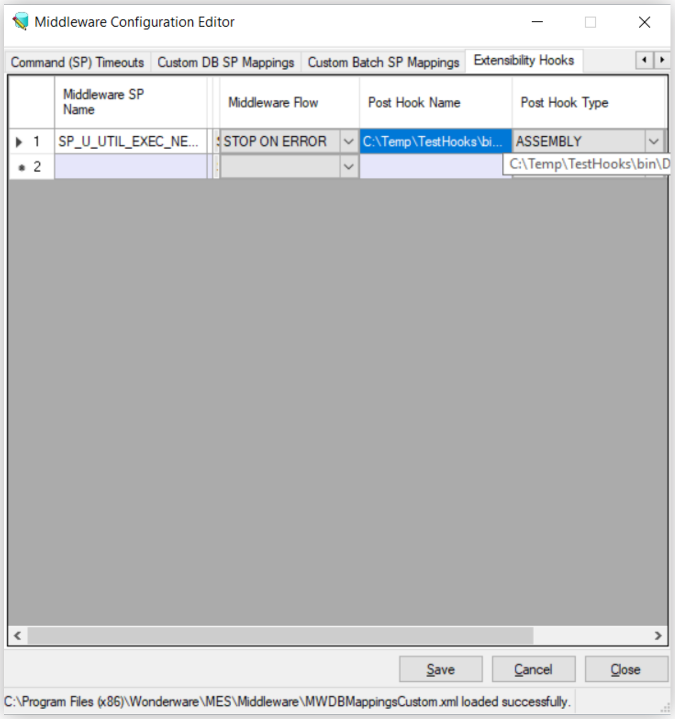
Once Within the Middleware Configuration Editor, a post hook is defined to call the assembly every time a new downtime is entered or updated 'SP_U_UTIL_EXEC_NEWREASON'. When the new downtime is entered / updated, the Event.txt is added /updated.

Once configured, the middleware server needs to be restarted before this event will be captured.

Copy the RunSimioPortalExperiment folder from the AVEVAMES.zip into the 'C:\Temp' folder. Modify the RunSimioPortalExperiment.exe.config.

Next, register the RunSimioPortalExperiment.exe as a windows service. Open a command prompt as an administrator (e.g. right click and select 'Run as Administrator').

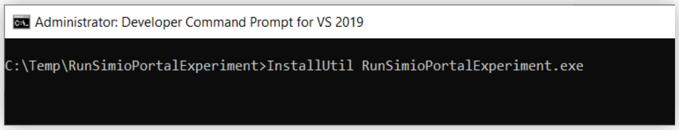
Navigate to 'C:\Temp\RunSimioPortalExperiment' From the command prompt, enter command 'InstallUtil RunSimioPortalExperiment.exe'



	Middleware SP Name	Middleware Flow	Post Hook Name	Post Hook Type
1	SP_U_UTIL_EXEC_NE...	STOP ON ERROR	C:\Temp\TestHooks\bi...	ASSEMBLY
2				

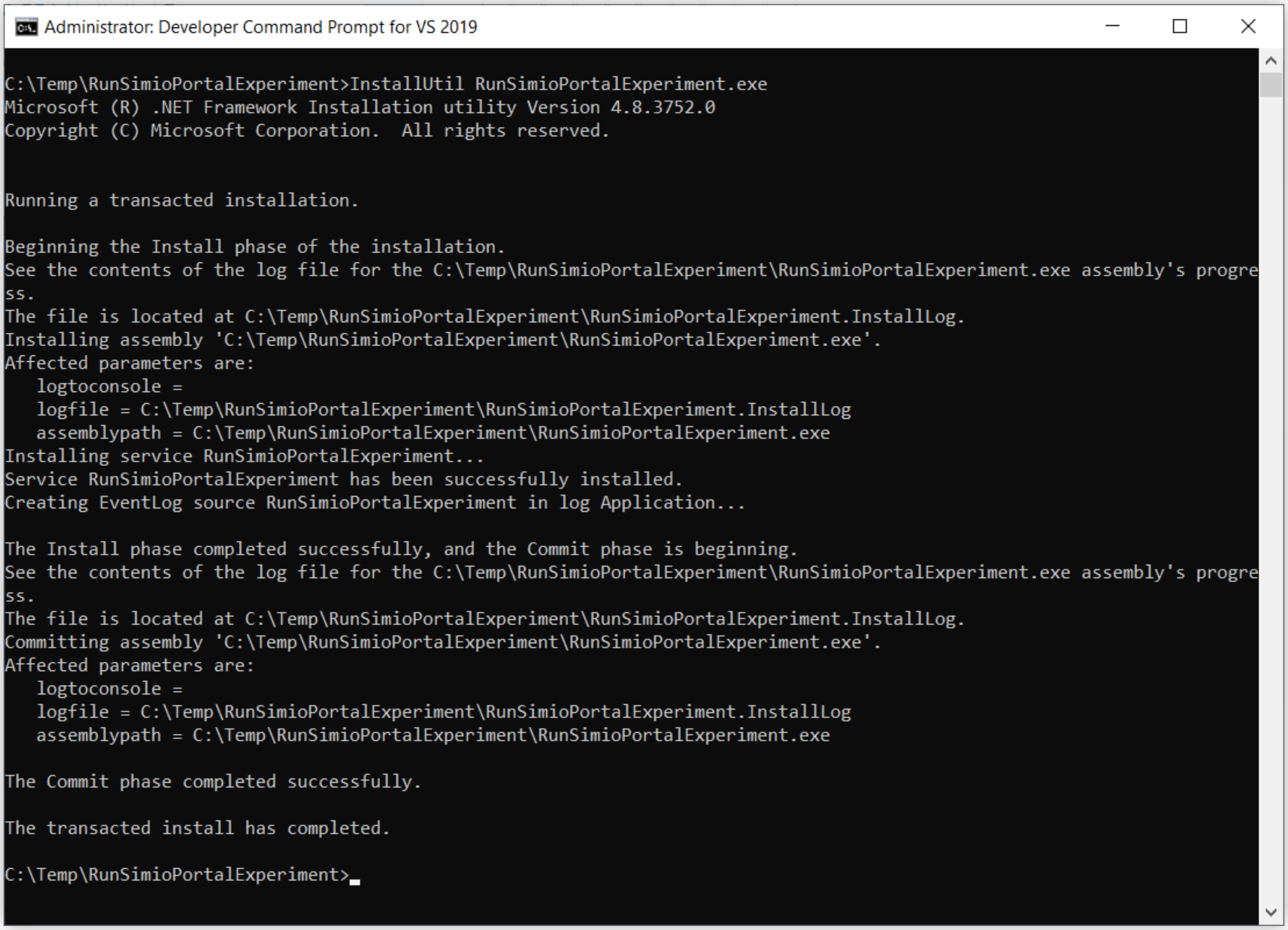
C:\Temp\TestHooks\bin\D

C:\Program Files (x86)\Wonderware\MES\Middleware\MWDBMappingsCustom.xml loaded successfully.



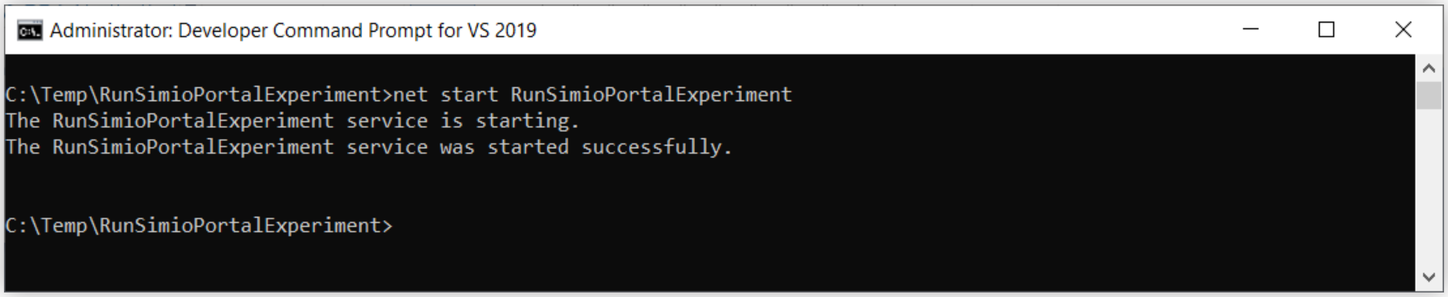
```
C:\Temp\RunSimioPortalExperiment>InstallUtil RunSimioPortalExperiment.exe
```

This will install the RunSimioPortalExperiment as a windows service.

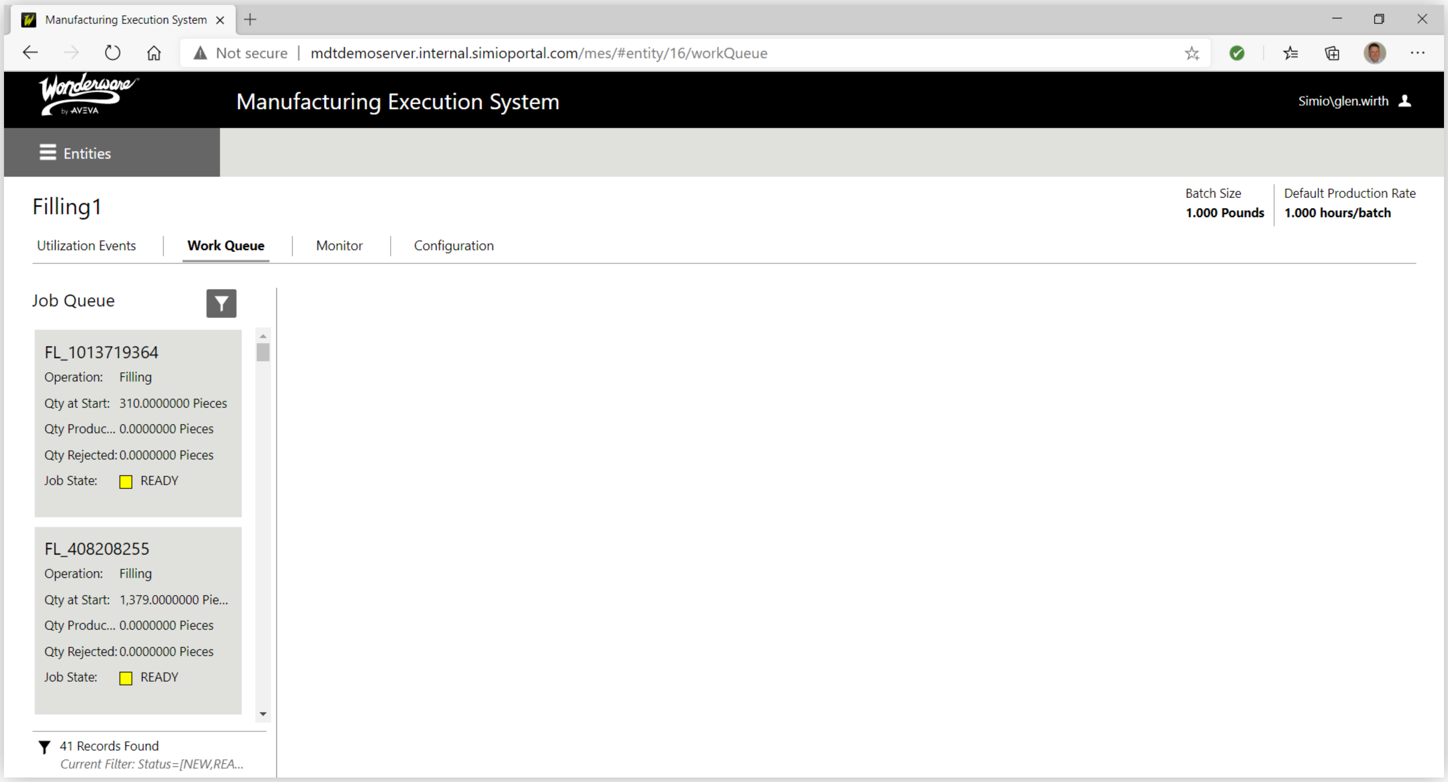


```
C:\Temp\RunSimioPortalExperiment>InstallUtil RunSimioPortalExperiment.exe
Microsoft (R) .NET Framework Installation utility Version 4.8.3752.0
Copyright (C) Microsoft Corporation.  All rights reserved.


Running a transacted installation.

Beginning the Install phase of the installation.
See the contents of the log file for the C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.exe assembly's progre
ss.
The file is located at C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.InstallLog.
Installing assembly 'C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.exe'.
Affected parameters are:
   logtoconsole =
   logfile = C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.InstallLog
   assemblypath = C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.exe
Installing service RunSimioPortalExperiment...
Service RunSimioPortalExperiment has been successfully installed.
Creating EventLog source RunSimioPortalExperiment in log Application...

The Install phase completed successfully, and the Commit phase is beginning.
See the contents of the log file for the C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.exe assembly's progre
ss.
The file is located at C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.InstallLog.
Committing assembly 'C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.exe'.
Affected parameters are:
   logtoconsole =
   logfile = C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.InstallLog
   assemblypath = C:\Temp\RunSimioPortalExperiment\RunSimioPortalExperiment.exe

The Commit phase completed successfully.

The transacted install has completed.

C:\Temp\RunSimioPortalExperiment>
```

Next, start the service. Enter 'net start RunSimioPortalExperiment' to start the service.



```
C:\Temp\RunSimioPortalExperiment>net start RunSimioPortalExperiment
The RunSimioPortalExperiment service is starting.
The RunSimioPortalExperiment service was started successfully.


C:\Temp\RunSimioPortalExperiment>
```


Manufacturing Execution System

Not secure | mdtdemoserver.internal.simioportal.com/mes/#entity/16/workQueue

Manufacturing Execution System

Simio\glen.wirth

Entities

Filling1

Batch Size	Default Production Rate
1.000 Pounds	1.000 hours/batch

Utilization Events | **Work Queue** | Monitor | Configuration

Job Queue

FL_1013719364

Operation:	Filling
Qty at Start:	310.0000000 Pieces
Qty Produc...	0.0000000 Pieces
Qty Rejected:	0.0000000 Pieces
Job State:	🟨 READY

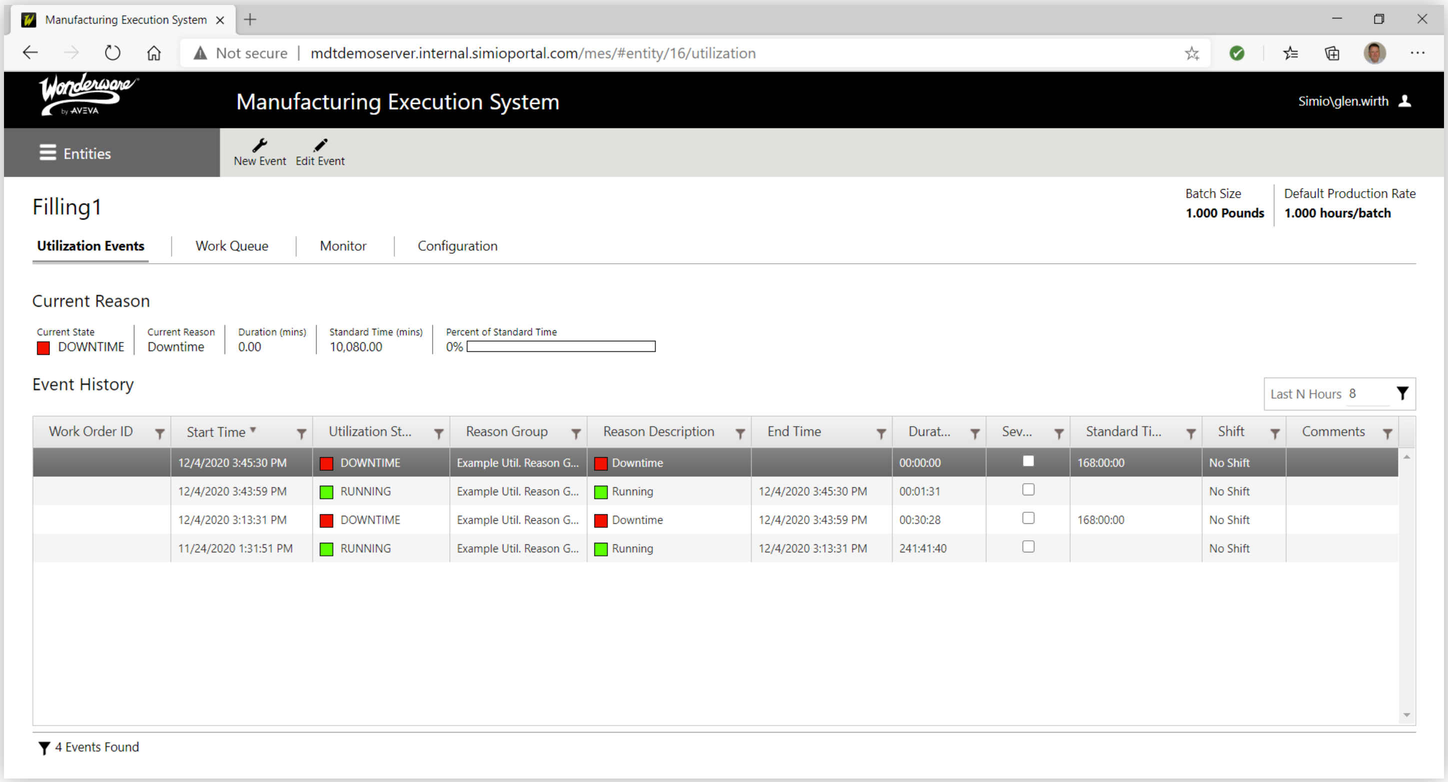
FL_408208255

Operation:	Filling
Qty at Start:	1,379.0000000 Pie...
Qty Produc...	0.0000000 Pieces
Qty Rejected:	0.0000000 Pieces
Job State:	🟨 READY

41 Records Found

Current Filter: Status=[NEW,REA...

To test the automatic generation of the schedule, check the work queue to verify that the current dispatch list on an entity.


Manufacturing Execution System

Not secure | mdtdemoserver.internal.simioportal.com/mes/#entity/16/utilization

Manufacturing Execution System

Simio\glen.wirth

Entities

New Event Edit Event

Filling1

Batch Size	Default Production Rate
1.000 Pounds	1.000 hours/batch

Utilization Events | Work Queue | Monitor | Configuration

Current Reason

Current State	Current Reason	Duration (mins)	Standard Time (mins)	Percent of Standard Time
🟥 DOWNTIME	Downtime	0.00	10,080.00	0%

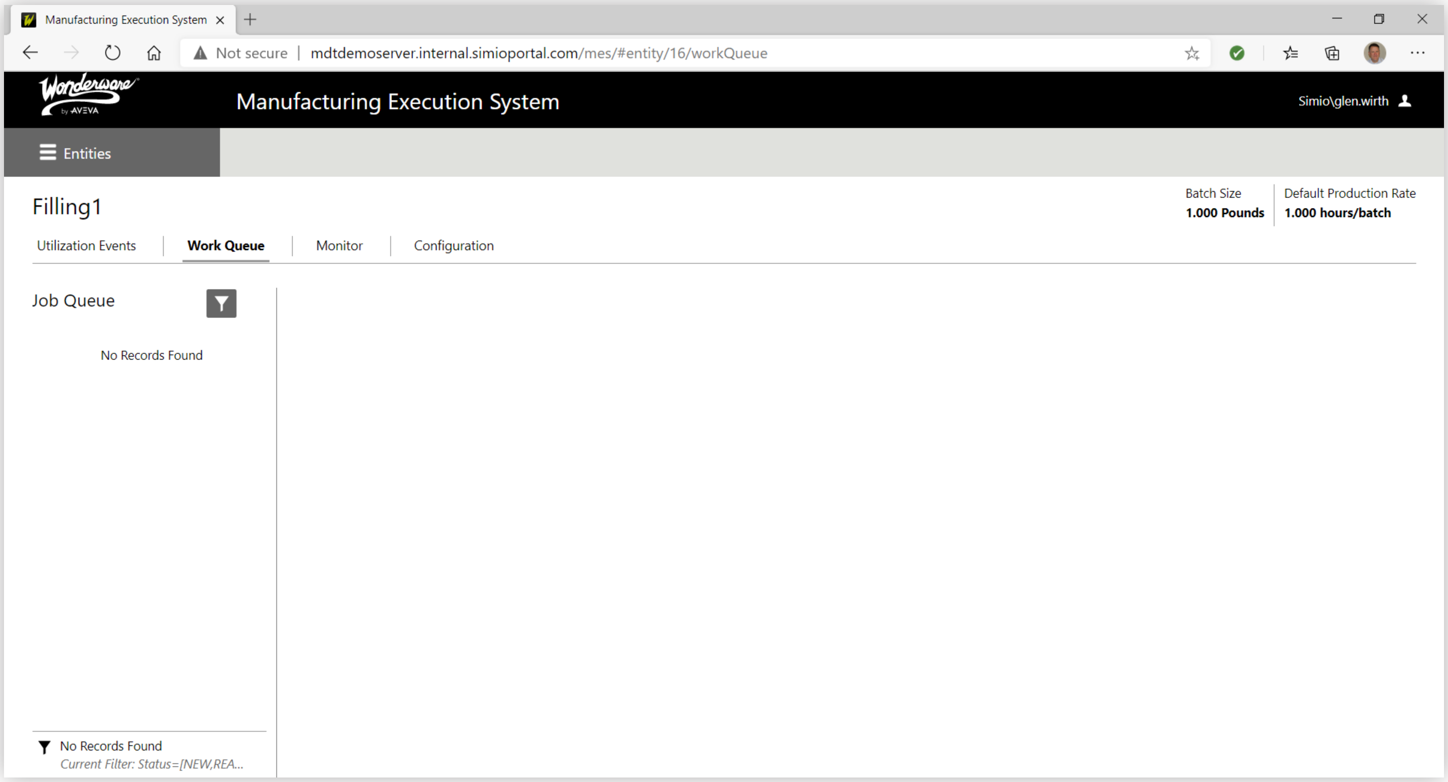
Event History

Last N Hours 8

Work Order ID	Start Time	Utilization St...	Reason Group	Reason Description	End Time	Durat...	Sev...	Standard Ti...	Shift	Comments
	12/4/2020 3:45:30 PM	🟥 DOWNTIME	Example Util. Reason G...	🟥 Downtime		00:00:00	☐	168:00:00	No Shift	
	12/4/2020 3:43:59 PM	🟩 RUNNING	Example Util. Reason G...	🟩 Running	12/4/2020 3:45:30 PM	00:01:31	☐		No Shift	
	12/4/2020 3:13:31 PM	🟥 DOWNTIME	Example Util. Reason G...	🟥 Downtime	12/4/2020 3:43:59 PM	00:30:28	☐	168:00:00	No Shift	
	11/24/2020 1:31:51 PM	🟩 RUNNING	Example Util. Reason G...	🟩 Running	12/4/2020 3:13:31 PM	241:41:40	☐		No Shift	

4 Events Found

Enter a downtime on the entity.


Manufacturing Execution System
Not secure | mdtdemoserver.internal.simioportal.com/mes/#entity/16/workQueue
Manufacturing Execution System
Simio\glen.wirth
Entities
Utilization Events
Work Queue
Monitor
Configuration
Job Queue

Filling1

Batch Size	Default Production Rate
1.000 Pounds	1.000 hours/batch

No Records Found

No Records Found
Current Filter: Status=[NEW,REA...

Verify that the schedule has changed.

20 Appendix B

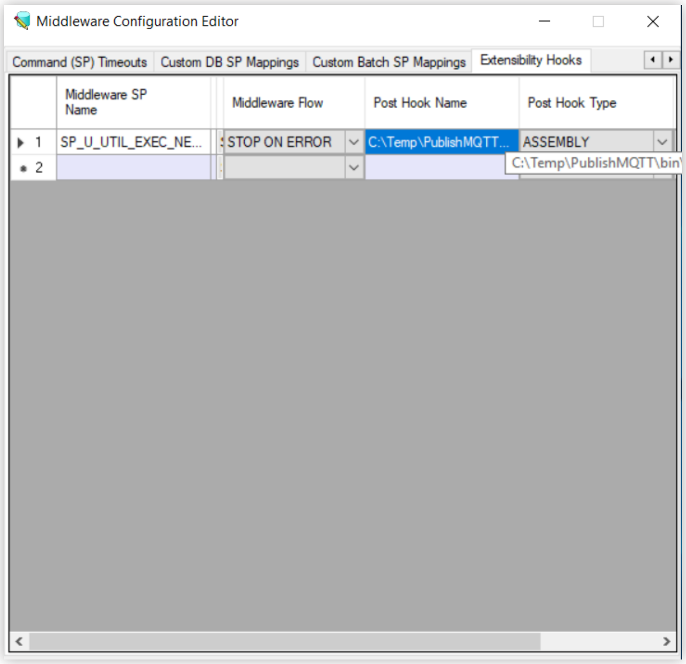
Automatic Reschedule Using PublishMQTT and Node-RED flow.

To automate reschedule, the AVEVA MES Middleware Extensibility Hooks and the Simio Portal Web API are used. The AVEVA MES Middleware Extensibility Hooks are used to capture the events in AVEVA MES. Once the events are captured, the Node-Red flow will see the event and automatically re-run the schedule using the Simio Portal Web API. After the scheduling run, the schedule will automatically be sent back to MES and the schedule will be published on Simio Portal.

Instead of the TestHooks (previous appendix), we are going to use PublishMQTT instead.

C:\Temp\PublishMQTT\bin\Debug\PublishMQTT.dll;PublishMQTTNamespace.PublishMQTTClass;PublishMQTTMethod

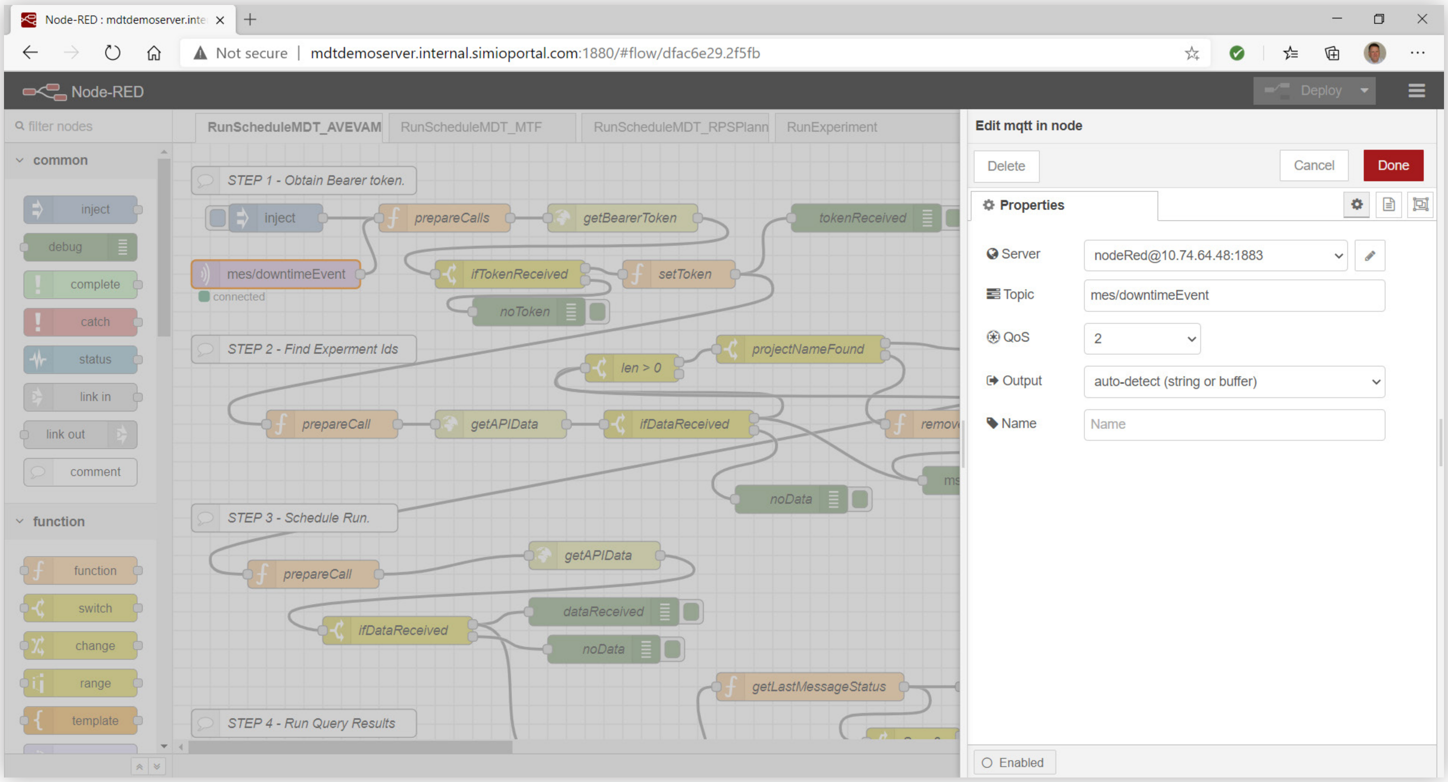
Once Within the Middleware Configuration Editor, PublishMQTT event is defined to call the assembly every time a new downtime is entered or updated 'SP_U_UTIL_EXEC_NEWREASON'. When the new downtime is entered / updated, the MQTT Topic 'mes/downtimeEvent' is notified.



	Middleware SP Name	Middleware Flow	Post Hook Name	Post Hook Type
1	SP_U_UTIL_EXEC_NE...	STOP ON ERROR	C:\Temp\PublishMQTT...	ASSEMBLY
2				

C:\Temp\PublishMQTT\bin\

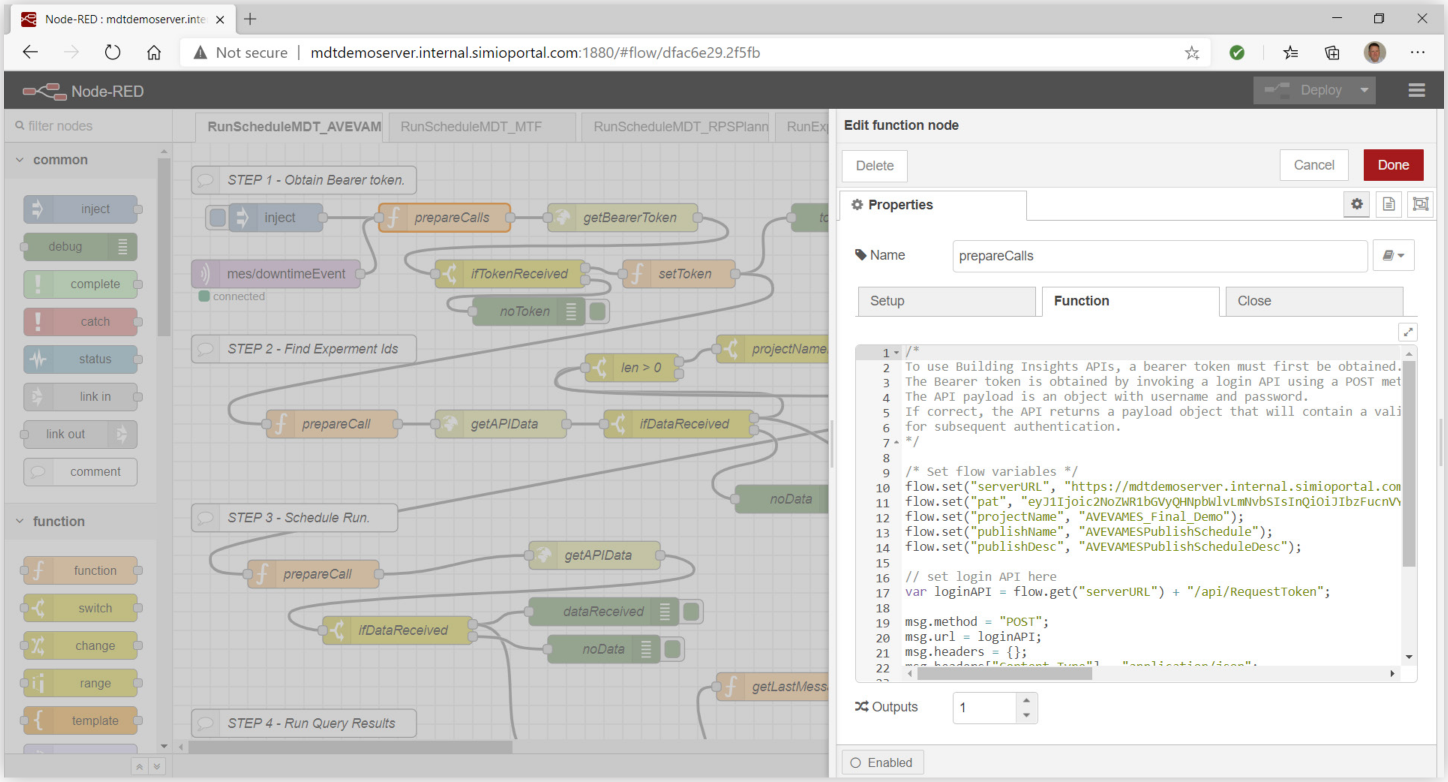
Once configured, the middleware server needs to be restarted before this event will be captured.



Edit mqtt in node

- Server: nodeRed@10.74.64.48:1883
- Topic: mes/downtimeEvent
- QoS: 2
- Output: auto-detect (string or buffer)
- Name: Name

Then a Node-RED flow is deployed to capture the event from MQTT. Once captured, the schedule is rerun and published on Simio portal.



Edit function node

- Name: prepareCalls
- Outputs: 1

```
/*
To use Building Insights APIs, a bearer token must first be obtained.
The Bearer token is obtained by invoking a login API using a POST met
The API payload is an object with username and password.
If correct, the API returns a payload object that will contain a vali
for subsequent authentication.
*/

/* Set flow variables */
flow.set("serverURL", "https://mdtdemoserver.internal.simioportal.com
flow.set("pat", "eyJ1Ijoic2NoZWR1bGVyQHNpbWlvLmNvbSIsInQiOiJIbzFucnVY
flow.set("projectName", "AVEVAMES_Final_Demo");
flow.set("publishName", "AVEVAMESPublishSchedule");
flow.set("publishDesc", "AVEVAMESPublishScheduleDesc");

// set login API here
var loginAPI = flow.get("serverURL") + "/api/RequestToken";

msg.method = "POST";
msg.url = loginAPI;
msg.headers = {};
```

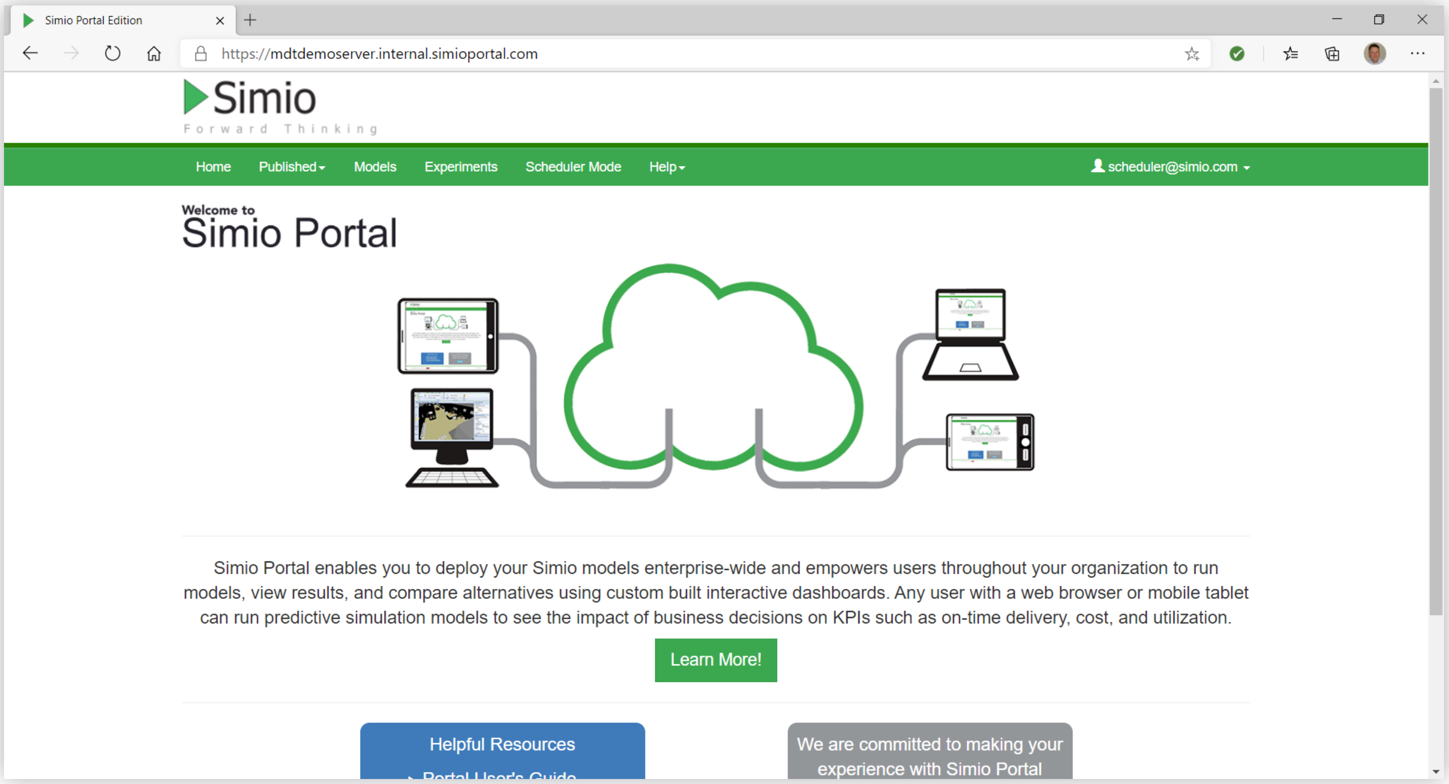
The first prepareCall function is used to configure the running and publishing of the schedule.

Once configured and deployed, the downtime works the same as in Appendix A.

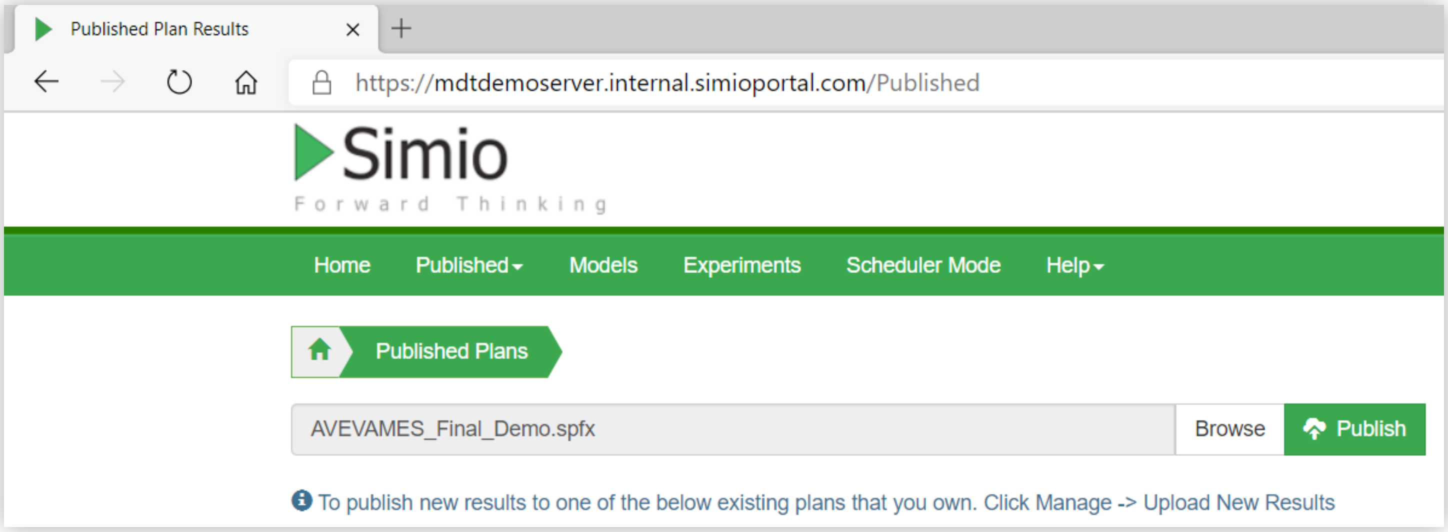
Publishing the Schedule Using the Simio Portal

An alternative method for publishing the Simio schedule uses the Simio Portal. The Simio Portal is a web based model repository used to publish the production schedule to be viewed throughout the organization. The Simio Portal can be hosted within an enterprise or hosted on Microsoft Azure.

In this example, we will publish the Simio schedule to Microsoft Azure. Once the user logs into the Simio portal, they will be provided a number of capabilities based on their permissions. This user has been given full access in the portal.



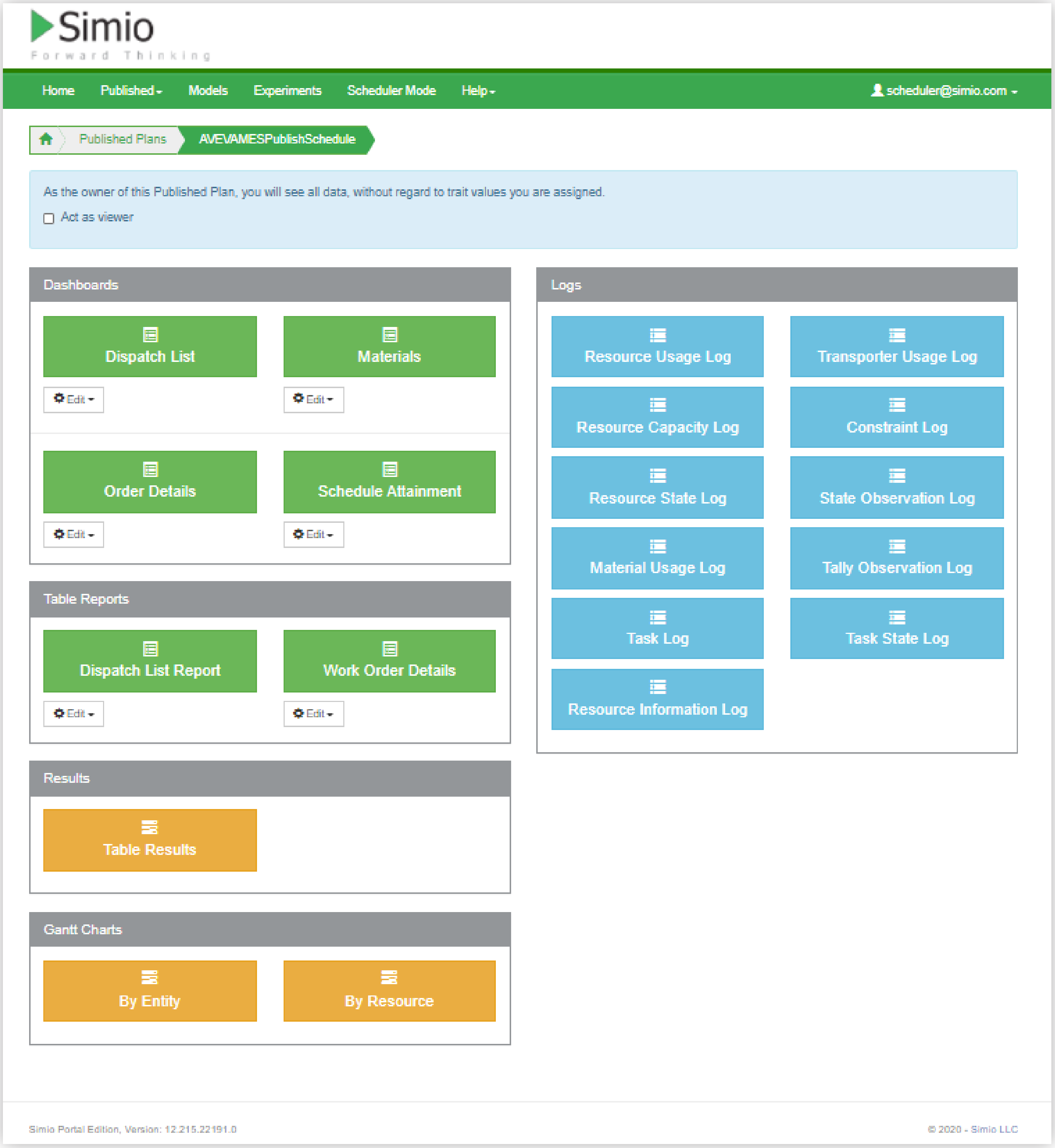
Simio Portal enables you to deploy your Simio models enterprise-wide and empowers users throughout your organization to run models, view results, and compare alternatives using custom built interactive dashboards. Any user with a web browser or mobile tablet can run predictive simulation models to see the impact of business decisions on KPIs such as on-time delivery, cost, and utilization.



AVEVAMES_Final_Demo.spfx

To publish new results to one of the below existing plans that you own. Click Manage -> Upload New Results

The user selects Published to upload a model.



Home Published Models Experiments Scheduler Mode Help scheduler@simio.com

Published Plans AVEVAMESPublishSchedule

As the owner of this Published Plan, you will see all data, without regard to trait values you are assigned.

Act as viewer

Dashboards: Dispatch List, Materials, Order Details, Schedule Attainment

Table Reports: Dispatch List Report, Work Order Details

Results: Table Results

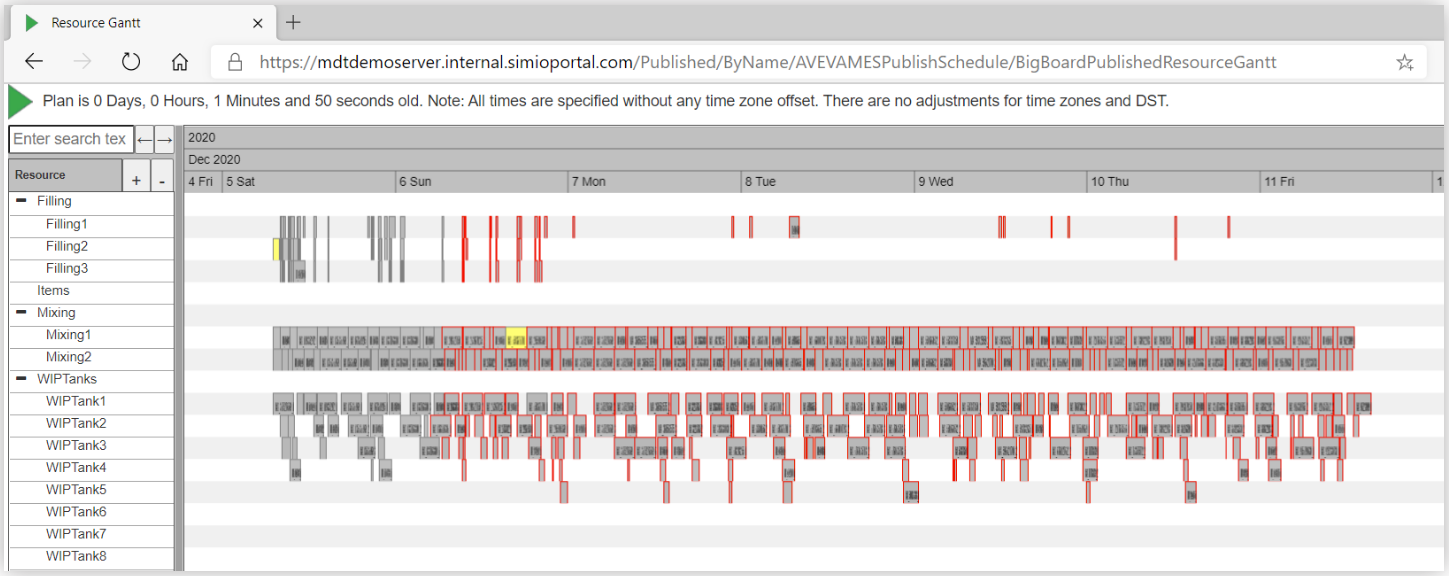
Gantt Charts: By Entity, By Resource

Logs: Resource Usage Log, Transporter Usage Log, Resource Capacity Log, Constraint Log, Resource State Log, State Observation Log, Material Usage Log, Tally Observation Log, Task Log, Task State Log, Resource Information Log

Simio Portal Edition, Version: 12.215.22191.0

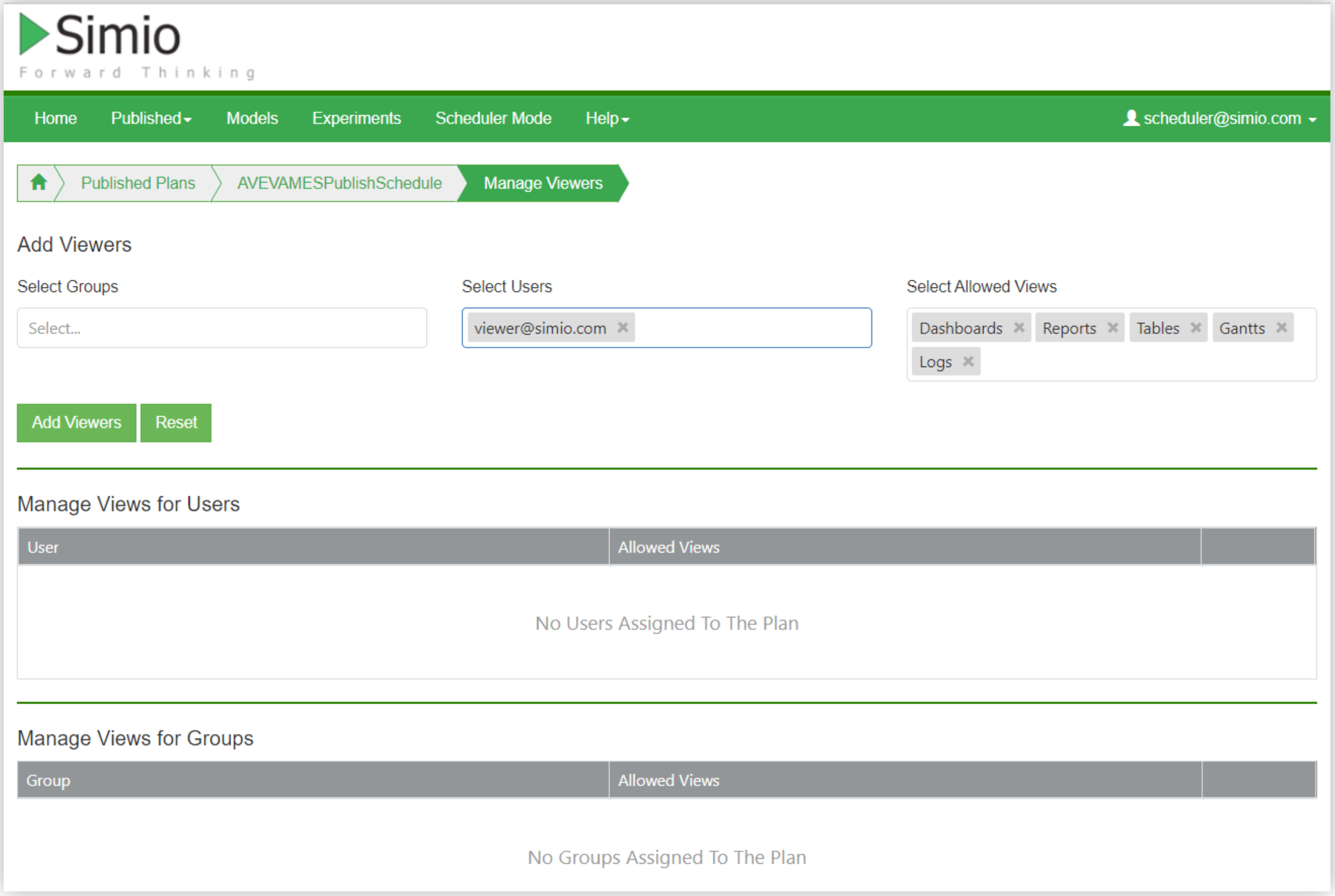
© 2020 - Simio LLC

Once the model is uploaded to the Portal, the user will have the option to display Dashboards, Table Reports, Results, Gantt Charts and Logs.



Plan is 0 Days, 0 Hours, 1 Minutes and 50 seconds old. Note: All times are specified without any time zone offset. There are no adjustments for time zones and DST.

Under Gantt Charts, select 'By Resource'. This will display the Resource Gantt chart



Add Viewers

Select Groups	Select Users	Select Allowed Views
Select...	viewer@simio.com	Dashboards, Reports, Tables, Gantts, Logs

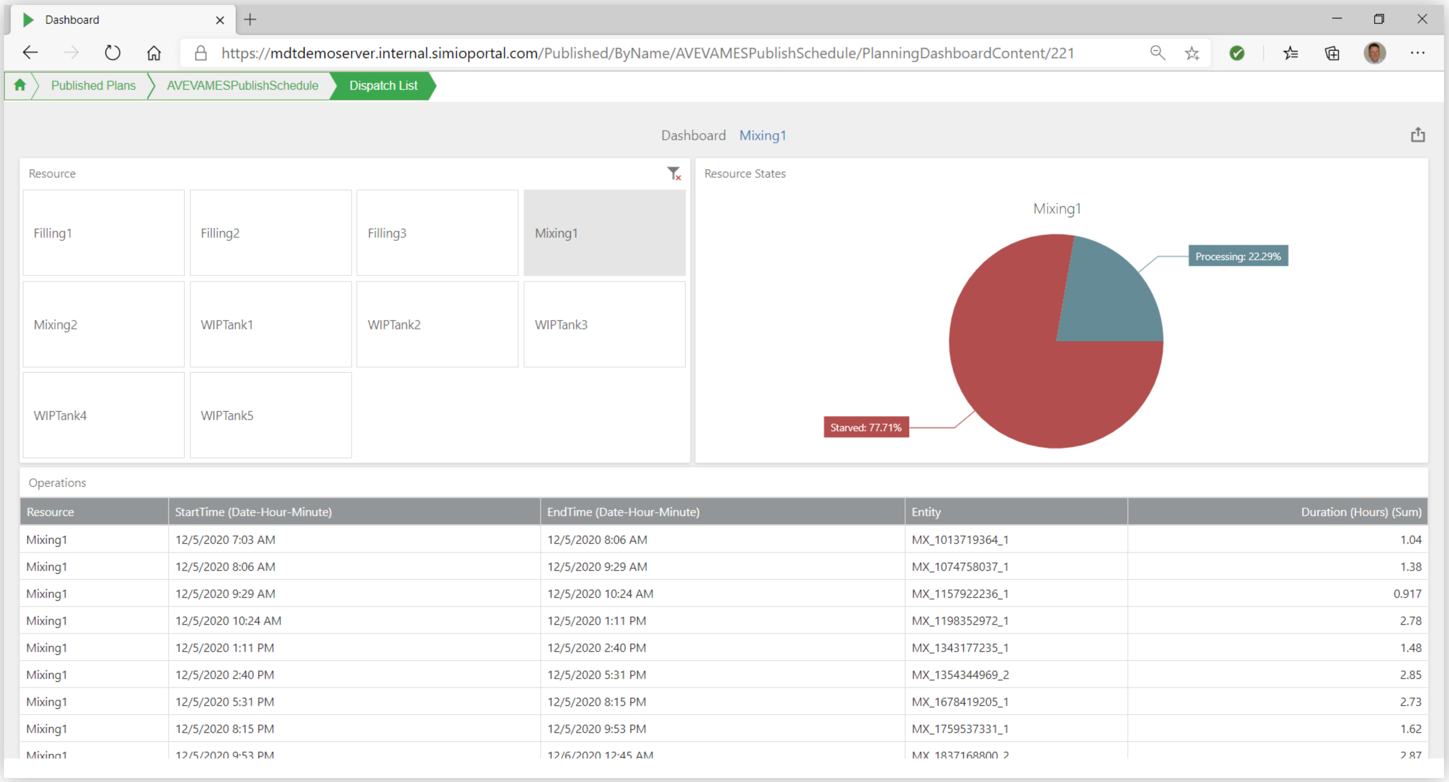
Manage Views for Users

User	Allowed Views	
No Users Assigned To The Plan		

Manage Views for Groups

Group	Allowed Views	
No Groups Assigned To The Plan		

Dashboards and Reports can be configured so only certain users can view them. When the viewer@simio.com user access the schedule, they only view the Dispatch List



Resource	StartTime (Date-Hour-Minute)	EndTime (Date-Hour-Minute)	Entity	Duration (Hours) (Sum)
Mixing1	12/5/2020 7:03 AM	12/5/2020 8:06 AM	MX_1013719364_1	1.04
Mixing1	12/5/2020 8:06 AM	12/5/2020 9:29 AM	MX_1074758037_1	1.38
Mixing1	12/5/2020 9:29 AM	12/5/2020 10:24 AM	MX_1157922236_1	0.917
Mixing1	12/5/2020 10:24 AM	12/5/2020 1:11 PM	MX_1198352972_1	2.78
Mixing1	12/5/2020 1:11 PM	12/5/2020 2:40 PM	MX_1343177235_1	1.48
Mixing1	12/5/2020 2:40 PM	12/5/2020 5:31 PM	MX_1354344969_2	2.85
Mixing1	12/5/2020 5:31 PM	12/5/2020 8:15 PM	MX_1678419205_1	2.73
Mixing1	12/5/2020 8:15 PM	12/5/2020 9:53 PM	MX_1759537331_1	1.62
Mixing1	12/5/2020 9:53 PM	12/6/2020 12:45 AM	MX_1837168800_2	2.87

Here is an example of the Dispatch List Dashboard provided to the viewer@simio.com user.

Simio

Forward Thinking

504 Beaver Street
Sewickley, PA 15143
Telephone: 412-528-1576
Fax: 412-253-9378
Email: info@simio.com
Web: www.simio.com